

Discrete modelling of bacterial conjugation dynamics

1

Angel Goñi-Moreno¹ and Martyn Amos¹

1 School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester M1 5GD, UK.

contact: Angel Goñi-Moreno: A.Moreno@mmu.ac.uk; Martyn Amos: M.Amos@mmu.ac.uk

Abstract

In bacterial populations, cells are able to cooperate in order to yield complex *collective* functionalities. Interest in population-level cellular behaviour is increasing, due to both our expanding knowledge of the underlying biological *principles*, and the growing range of possible *applications* for engineered microbial consortia. Researchers in the field of synthetic biology - the application of engineering principles to living systems - have, for example, recently shown how useful decision-making circuits may be *distributed* across a bacterial population. The ability of cells to interact through small signalling molecules (a mechanism known as *quorum sensing*) is the basis for the majority of existing engineered systems. However, horizontal gene transfer (or *conjugation*) offers the possibility of cells exchanging messages (using DNA) that are much more information-rich. The potential of engineering this conjugation mechanism to suit specific goals will guide future developments in this area. Motivated by a lack of computational models for examining the specific dynamics of conjugation, we present a simulation framework for its further study. We present an agent-based model for conjugation dynamics, with realistic handling of physical forces. Our framework combines the management of intercellular interactions together with simulation of intracellular genetic networks, to provide a general-purpose platform. We validate our simulations against existing experimental data, and then demonstrate how the emergent mixing patterns of multi-strain populations can affect conjugation dynamics. Our model of conjugation, based on a probability distribution, may be easily tuned to correspond to the behaviour of different cell types. Simulation code and movies are available at <http://code.google.com/p/discus/>.

1 Introduction

Researchers in the emerging field of synthetic biology [1] have demonstrated the successful construction of devices based on populations of engineered microbes [2, 3, 4]. Recent work has focussed attention on the combination of single-cell intracellular devices [5, 6] with intercellular engineering, in order to build increasingly complex systems.

To date, most work on engineered cell-cell communication has focussed on quorum-sensing (QS) [7], which may be thought of as a communication protocol to facilitate inter-bacterial communication via the generation and receiving of small signal molecules. However, recent studies on DNA messaging [8] highlight the importance and utility of transferring whole sets of DNA molecules from one cell (the so-called donor) to another (the recipient). Bacterial *conjugation* is a cell-to-cell communication mechanism [9, 10] that enables such transfers to occur.

In this paper we present a simulation platform that realistically simulates (in a modular fashion)

both intracellular genetic networks and intercellular communication via conjugation. To our knowledge, this is the first such platform to offer both of these facilities. We first review previous work on cell simulation, before presenting the details of our model. We validate it against previous experimental work, and then discuss possible applications of our method.

2 Previous work

The rapid development of bacterial-based devices is accompanied by a need for computational simulations and mathematical modelling to facilitate the characterisation and design of such systems. A number of platforms and methods are available for this purpose. Agent-based models (AbMs) are widely used [11], and were first used to study microbial growth in *BacSim* [12]. Continuous models have also been proposed [13], and recent developments make use of hardware optimisation, by using GPUs (Graphics Processing Units) in order to scale

up the number of cells simulated [14].

Because of the complexity of the system under study, several computational platforms focus on either specific cellular behaviours (e.g., bacterial chemotaxis [15], morphogenesis of dense tissue like systems [16]) , or on specific organisms (e.g., *Myxococcus xanthus* [17]). Platforms that incorporate cell-cell communication generally focus their attention on quorum-sensing. Simulations of conjugation do exist, but these consider cells as abstracted *circular objects* [18, 19]. We demonstrate in this paper how a consideration of the *shape* of cells is an essential feature for understanding the conjugation behaviour of the population. We now describe our model for bacterial growth, in which conjugation is handled explicitly.

3 Methods

We use an *individual-based modelling* approach [20] to the study of conjugation dynamics. This models each cell as an individual, mobile entity, each of which is subject to physical forces arising from contact with other cells and the environment (e.g., surfaces). Each cell has a number of different *attributes*, listed in Table 1, which correspond to various physiological states and characteristics.

Bacteria are modelled as rod-shaped cells with a constant radius (parameter `width` in Table 2). Elongation processes occur along the longitudinal axis, which has a minimum dimension of `length`, and division takes place whenever a the cell measures $2 * \text{length}$. The division of a cell into two new daughter cells is also controlled by `max_overlap`, which monitors the physical *pressure* affecting each cell; if the pressure exceeds this parameter value, the cell delays its growth and division. Thus, a cell with pressure grows slower than without it. In Figure 1A we see an snapshot of a population with different cell lengths, due to the pressure-dependent behaviour. The global parameter `growth_speed` (Table 2) also helps us simulate cell flexibility in a realistic fashion. This parameter defines a “cut off” value for the number of iterations in which the physics engine must resolve *all* the current forces and collisions. Thus, smaller values will cause the solver to be effectively “overloaded”, and some colli-

sions may, as a result, be partially undetected. This² means that cells behave as flexible shapes, which gives the simulation a more realistic performance. In Figure 1C we show how changes in `growth_speed` affect the simulation, using bigger (left) to smaller (right) values.

Horizontal genetic transfer (or conjugation) is modelled using an *elastic spring* to connect donor and recipient cells. Parameter `c_time` defines the duration of that linkage, which determines the time in which the DNA is transferred. The springs are constantly monitored to ensure that they physically connect both cells during conjugation. Importantly, during conjugation, the resolution of collisions involving relevant cells considers the forces produced by the spring connection, in order to calculate the final movement of the bacteria. By coupling cells in this way, we obtain realistic population-level physical patterns that emerge as a result of large numbers of conjugation events. Figure 1B shows this process, with a donor cell (red) and a recipient cell (grey) which becomes a transconjugant (yellow). A transconjugant cell is one that was initially a recipient, but which has been conjugated during its lifetime. Thus, it already has the DNA information transferred by the donor.

This agent-based algorithm has an iteration-driven structure, where - after initialisation of the main global parameters - it repeatedly performs the following steps for each cell:

1. Update springs (position and timing).
2. Perform cell division (if cell is ready).
3. Elongate cell (every `growth_speed` steps).
4. Handle conjugation.
5. Update physical position.

Conjugation decisions (step 4) made by cells are driven by three sequential steps:

1. The cell *decides*, following a probability distribution, whether or not to conjugate (one trial per iteration).
2. If conjugating, randomly select a mate from surrounding bacteria (if present).

Table 1. Cell attributes.

Attribute	type	Definition
shape	<i>pymunk</i> .Shape	Shape of the cell
program	$[m_0 \dots m_i]$	List of the i regulatory network molecules (m)
elongation	[int,int]	Elongation values (one per cell pole)
position	[x,y]	Coordinates of centre point, x and y
speed	float	Velocity
conjugating	Boolean	Conjugation state
plasmid	Boolean	Program state (present/not present)
role	int	Donor (0), recipient (1) or transconjugant (2)
partner	int	Role of plasmid transfer cell

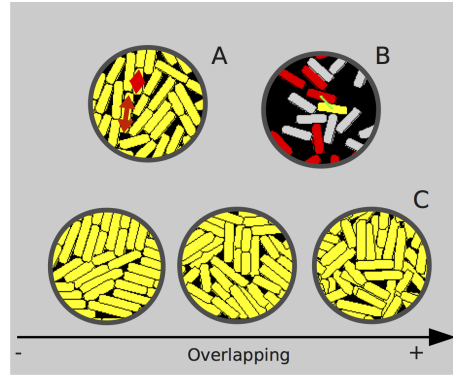


Figure 1. Cell behaviours at low scale. (A): Different cell length due to asynchronous growth (pressure dependent). Two cells marked with red arrows. (B): A donor cell (red) starts the conjugation process with a recipient (grey) which turns into transconjugant (yellow). The pilus (green) is an elastic spring that links the two cells until the process is finished. (C): Different overlapping levels within the cells of a population.

3. If valid mate is found, effect conjugation transfer.

The discrete probability distribution used for the conjugation process is $C(N, p, \text{c_time})$, where N is the number of trials in a cell lifetime ($\text{width} * \text{length}$), p is the success probability in each trial (with $p \in [0..1]$) and c_time is the time interval during $p = 0.0$ (i.e., when the cell is already conjugating). As stated in [21], p can vary, depending on whether the cell is a donor (p_d), a transconjugant that received the DNA message from a donor (p_t1), or a transconjugant that received the DNA from another transconjugant (p_t2).

Intracellular circuitry is modelled separately, and then *introduced* into each cell by storing the state of the circuit in an attribute of the cell (**program**). Thus, there are effectively as many copies of the circuit as cells in the simulation. This circuit simulation is implemented in a modular fashion, so that the internal cellular “program” may be easily replaced with any other. In this paper we demonstrate the principle using a two-component genetic oscillator as the DNA message that is exchanged through conjugation. The ordinary differential equations (ODEs) for this circuit are:

$$\frac{dx}{dt} = \Delta \left(\beta \frac{1 + \alpha x^2}{1 + x^2 + \sigma y^2} - x \right) \quad (1)$$

$$\frac{dy}{dt} = \Delta \gamma \frac{1 + \alpha x^2}{1 + x^2} - y \quad (2)$$

which are detailed in [22], as well as the meaning and value of the parameters (we use the same values in the code provided). We have also recently used our software platform to investigate the spatial behaviour of a *reconfigurable* genetic logic circuit [23], which demonstrates how it may easily be modified to accommodate different sets of equations. The actions controlling the growth rates of cells occur on a longer time scale than the integration steps that control molecular reactions (as equations 1 and 2). In order to ensure synchronisation, the parameter **network_steps** defines the number of integration steps of the ODEs that run per **Gt**. Thus, a number of **network_steps/Gt** integration steps will update the attribute **network** of each cell every iteration.

Other important physical parameters listed in Table 2 are **spring_rest_length**, **spring_stiffness** and

spring_damping; these are three parameters to model the material and behaviour of the bacterial pilus (i.e. the spring) during conjugation. Parameter **cell_infancy** is a delay period, during which a cell is considered to be too young to conjugate (as observed experimentally [21]). Parameters **pymunk_steps** and **pymunk_clock_ticks** are used by the physics engine to update the world, and may be adjusted by the user in order to alter the performance of the simulation (machine dependent). Parameters **bac_mass** and **bac_friction** play a role in collision handling.

Our platform is written in *Python*, and makes use of the physics engine *pymunk* (www.pymunk.org) as a wrapper for the 2D physics library *Chipmunk*, which is written in C (www.chipmunk-physics.net/). As cells are represented as semi-rigid bodies in a 2D lattice, *pymunk* handles the physical environment on our behalf. For monitoring purposes, parameters **Gt** and **real_Gt** allow us to stabilise the relation between iterations and clock minutes: $\text{minute} = \text{Gt}/\text{real_Gt}$ (units: iterations). The platform, which we call DiSCUS (Discrete Simulation of Conjugation Using Springs) is available at the project repository at <http://code.google.com/p/discus/>.

4 Results

We now describe the results of experiments to validate our conjugation model, using four sets of simulations. As we aim to understand the behaviour of cells in small-scale two-dimensional populations (as occur in microfluidic environments), we avoid the sorts of extreme overlapping situations shown in Figure 1C(right). We first validate individual conjugation dynamics; then we validate the biomechanical properties of the simulation; the third set of experiments concerns the transfer of the two-component oscillator, and the final set of experiments study the effects of mixing on conjugation dynamics. We now describe in detail the results of each set of experiments.

4.1 Conjugation dynamics

The objective of the first set of experiments is to validate the software in terms of *conjugation dynamics*. For that purpose, we first focus on conjugation,

Table 2. Global simulation parameters.

Parameter	Definition	Parameter	Definition
screenview	Size of the simulated <i>world</i>	network_steps	Number of steps of the ODEs per Gt
max_overlap	Pressure tolerance of cells	number_donors	Initial number of donor cells
width	Width of each cell (lattice squares)	number_recipients	Initial number of recipient cells
length	Length of each cell (lattice squares)	spring_rest_length	Natural spring expansion/contraction
growth_speed	Iterations between elongation processes	spring_stiffness	The tensile modulus of the spring
Gt	Doubling time of the simulated cells (iterations)	spring_damping	The amount of viscous damping to apply
real_Gt	Real doubling time of the studied cells (minutes)	cell_infancy	Time lag (percentage)
p.d	Probability of conjugation event (donors)	pymunk_steps	Update the space for the given time step
p.t1	Probability of conjugation event (transconj.1)	pymunk_clock_ticks	Frame frequency (FPS - frames per second)
p.t2	Probability of conjugation event (transconj.2)	bac_mass	Mass of the cell (for calculating the moment)
c.time	Duration of the conjugation process	bac_friction	Friction coefficient (Coulomb friction model)

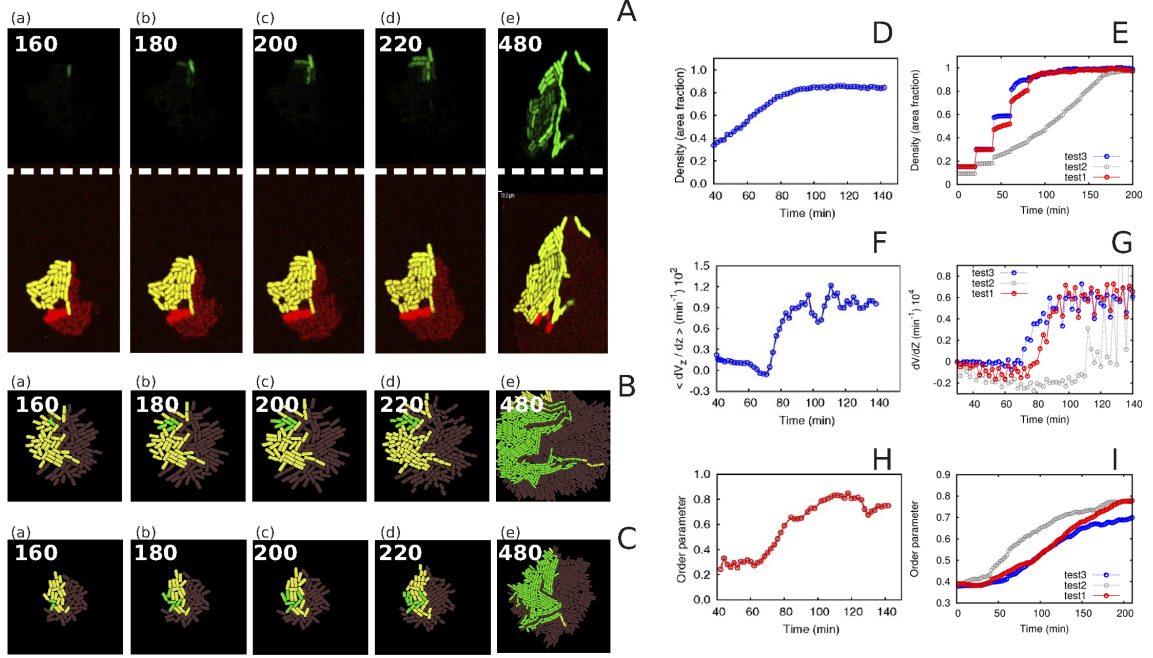


Figure 2. Validation of cell movement and conjugation dynamics using real data. (A): Figure extracted from [21] where a colony of *Pseudomonas putida* is divided into dark red donor cells (DsRed), yellow recipient cells (YFP) and transconjugants, expressing both yellow and green light (YFP and GFP). The upper row shows the transconjugant signal, and the bottom row shows the whole community. (B and C): Simulation results. Two simulations of similar colonies are recorded over exactly the same time intervals (min). The colours of the cells match the colours observed in (A). Graphs (D), (F) and (H) are extracted from [24], and show experimental results of *Escherichia coli* growth regarding density, velocity and ordering (respectively). Graphs (E), (G) and (I) correspond to simulations in similar conditions to [24], for the same parameters (density, velocity and ordering respectively). Tests 1, 2 and 3 in graphs correspond to different spatial distribution of cells inside the microfluidic channel (details in text).

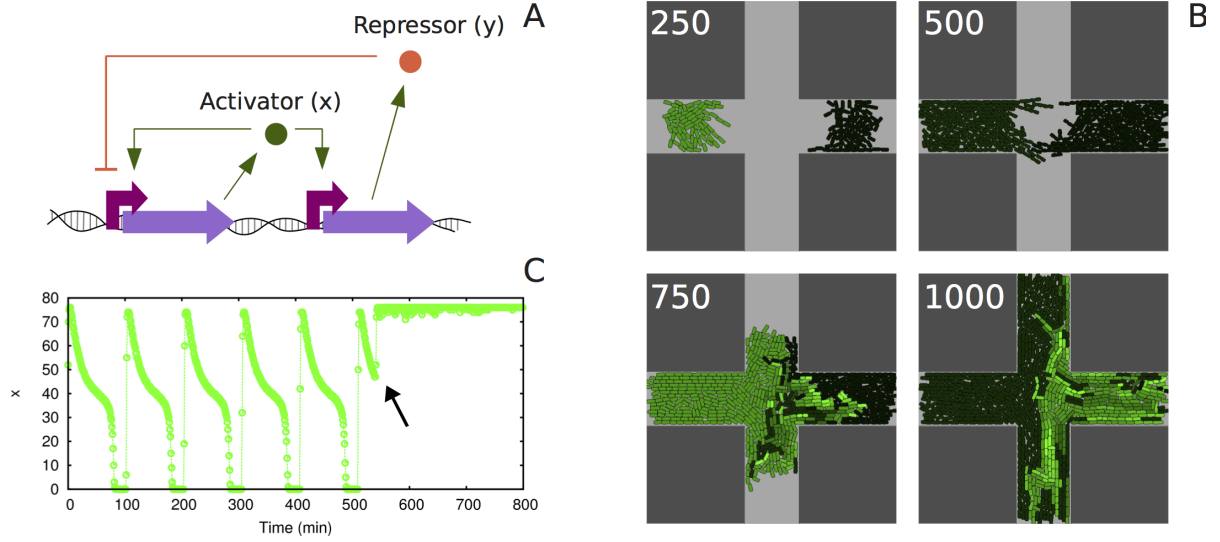


Figure 3. Horizontal transfer of a two-component genetic oscillator. A: Transcriptional-level design. The activator (green) acts on itself and on the repressor (red) by inducing the transcription of both. The repressor acts on the activator by repressing its transcription. As a result, molecule x (as well as y) oscillates in time. B: Cells growing in a cross-shaped channel. At 250 (minutes) we clearly see the position of both donor (left hand, with the oscillator inside) and recipient (right hand, empty) strains. The intensity of green colour denotes the amount of molecule x inside cells in that specific time. Through conjugation, the oscillator is *copied* to the initial recipient strain ($t = 1000$ min). C: Using the same experiment as in B, measurement over time of the maximum level of molecule x in a single cell. Black arrow highlights the point in time when conjugation starts ($t \simeq 550$ min).

using images of a *Pseudomonas putida* population (Figure 2A) extracted with permission from [21]. These show donor cells (dark red) growing in contact with recipients (yellow). The DNA information they share after conjugation makes the transconjugant cells display GFP (green fluorescent protein). We adjusted the parameters of our simulations until the behaviour matched the images of real cells (two simulations shown: Figures 2B and 2C), in terms of both time-series behaviour and the type of physical pattern displayed. It is important to note that the differential probabilities of conjugation of donors and transconjugants (higher in the latter) causes directional spreading of the DNA information. After the first transconjugant appears (160 minutes), the newly-formed transconjugants appear -most probably - in the immediate neighbourhood. The final parameter values used to reproduce this experiment are: `width=5`, `length=15`, `growth_speed=30`, `p_d=0.001`, `p_t1=0.02`, `p_t2=0.05` and `c_time=450` (the rest of the parameters are as defined in the DiSCUS distribution). Movie *DemoConjugation1* (found in the project repository) shows a simulation of a similar experiment where the transconjugants do not act as new donors.

4.2 Biomechanical properties

The second set of validation experiments focuses on *biomechanical movement*. We use data from [24], which describe an *Escherichia coli* colony growing in a microfluidic channel ($30 * 50 * 1 \mu\text{m}^3$) (Figures 2D, 2F and 2H). Using the same parameter setup (`width=5`, `length=24`, `growth_speed=30`) we highlight how different initial positioning of cells inside the channel can affect the final result (*test1*, with more cells observed in the centre than at the edges; *test2* with all cells initially in the centre; *test3* with all cells homogeneously spread along the channel). Density graphs (Figures 2D and 2E) show the increasing curve as the channel becomes more populated (results vary depending on which area is considered for monitoring). Velocity gradients (Figures 2F and 2G) depict the differential velocity across the longitudinal axis of the channel with respect to the centre (we see negative values when the cells in the centre move faster than the rest). The difference in the *y* axis is due to our considering different spacial intervals in the velocity gradient calculation. Ordering graphs (Figures 2H and 2I) are based on

calculating the cosine of a cell’s angle with respect to the longitudinal axis of the channel (e.g. angle 0, $\cos(0)=1$, completely aligned). As time increases, we see that the cells tend to align themselves.

4.3 Internal cell “program”

Figure 3 shows the results of the next set of experiments, aiming at studying the horizontal transfer of a *two-component oscillator*. The transcriptional-level design of the circuit is shown in Figure 3(A), which causes the molecular concentration of a repressor (*y*) and an activator (*x*) to oscillate over time. Each molecule is produced by a gene when its upstream promoter is activated. The activator can induce its own production at the same time as inducing the production of the repressor, which in turns inhibits the production of the activator. In Figure 3B we place (250 minutes) an initial donor colony (with the two-component oscillator inside) on the left-hand side of a cross-shaped channel, while a recipient colony is placed on the right.

As the equations for the oscillator have no stochasticity, every cell of the donor strain shows exactly the same state of the circuit as every other cell. At the beginning (250 minutes), these cells show a green colour (corresponding to the molecular concentration of *x*) which is switched off during the time intervals in which the repressor is *on* (see time profile of Figure 3C). When conjugation starts (at around $t \simeq 550$), the newly formed transconjugant cells are given the circuit but, importantly, they do not share the state of the circuit of the cells from which they receive the message. During the DNA transfer, it is only the plasmid (circuit *carrier*) that is copied into the recipient; therefore, both molecular concentrations are null, and the circuit begins its functioning from the initial stage. That is why we clearly observe different green intensities within the community. This asynchronous behaviour happens only in the transconjugants, while the circuits inside the donors always run synchronously (due to deterministic equations).

A time profile of the previous experiment is shown in Figure 3C, where the maximum level of activator (*x*) concentration in a single cell (compared with the whole population) is recorded over time. Before conjugation starts, all cells in the consortia display perfect synchrony. After conjugation (shown with an arrow on the graph) there is al-

ways a cell with the maximum level of activator, which demonstrates high asynchrony. All parameter values regarding cell dimensions or conjugation probabilities are the same as in Figures 2B and 2C. Parameters relevant to the oscillator are: `network_steps=18` and `Gt=450`. Movie *DemoConjugation2* (found in the project repository) shows this experiment and Movie *DemoDynamics1* shows donor cells growing with a stochastic version of the oscillator.

4.4 Effects of mixing

Conjugation behaviour within a population may be altered in different ways to achieve different behaviours, depending on the desired application. For example, in the previous experiments described in this paper, transconjugants are unable to act as recipients (simulating a *radical* entry exclusion [25]). That is to say, they will not receive more plasmids (genetic circuits) from either donors or transconjugants. Furthermore, we may also engineer the transconjugants to stop acting as *new donors* [10], so that only the original donors have the ability to transfer the DNA message. *Mixing* of the cell population becomes essential in this last scenario, in order to ensure maximal contact between donors and recipients. In Figure 3B we see how, at the end of the experiment (1000 minutes), donors and the transconjugants cover different areas of the channel (left and right respectively), without being mixed.

We now study the autonomous mixing behaviour of cells under different environmental conditions, with the third sets of experiments (Figure 4). Firstly we investigate how morphological changes in a longitudinal microfluidic channel can affect the patterns being formed by the consortia and its mixing. Figure 4A shows three bacterial strains (each shown in a different colour) growing in a channel from different starting points. As we can see, their mixing is highly improbable. The main reason for this is the velocity and directionality of the cells. As the cells are washed out at the edges of the channel, all of them *travel* (they only have passive movement while being pushed) at variable speed from centre to left or from centre to right. This causes the cells to have the same direction (see vector field), which in turn makes mixing more difficult. In Figures 4B and 4C we show the result of altering the morphology of the channels, by adding columns and zig-zag

walls, respectively. As a result, the cells show different directionality (see vector fields) and the strains have a higher probability of being mixed. In both experiments (unlike 4(A)), the three strains are in contact at some point. If the experimental application relied on the conjugation of purple and yellow cell pairs, for example, we can see that these physical changes in the channel would be essential.

Another way to intensify the mixing of strains in a microfluidic trap is to change the main channel *flow strength*, with the objective of creating turbulence inside the trap. Figures 4D and 4E show a three-strain population growing in a trap (identical initial positioning of cells in both) with the only difference being that the strong flow in the main channel (white arrows) of 4E creates turbulence (in the direction of the circled arrow). Two different colonies are simulated in each experiment, inside both symmetrical and independent traps. We see how turbulence helps the cells to get mixed, thanks to the constant change of direction they display (see vector field in Figure 4F). Furthermore, we can avoid *missing* one strain, as happens with the yellow cells in 4D (see Movie *DemoDynamics2* -found in the project repository-), where another run of this experiment is shown).

Investigations of how manual mixing can affect conjugation frequencies are described in in [10], using an *Escherichia coli* population. We now reproduce those results using our software, and give valuable insight into the reasons for that behaviour: the *isolation* of the recipients. For that purpose (Figure 5) we grow a population of donors (D, red) and recipients (R, yellow) in which the ratio D/R is 50% and the transconjugants (T, green) are unable to act as new donors. The frequency of conjugation, Y , is measured as $Y = T/(R + T)$. The graph in Figure 5C shows the frequency after 560 minutes of *untouched* populations (not mixed, blue bars) and populations that have been *manually mixed* at 420 minutes (red bars). The difference that the mixing produces is based on the isolation of the recipients in untouched populations. Figure 5A shows two different occasions in which clusters of recipients are formed, where the transconjugants do not allow donors to reach new possible mates. After the population is completely “shuffled” (5B), the clusters are dissolved, and new pairs of donor-recipient can arise in the new topology.

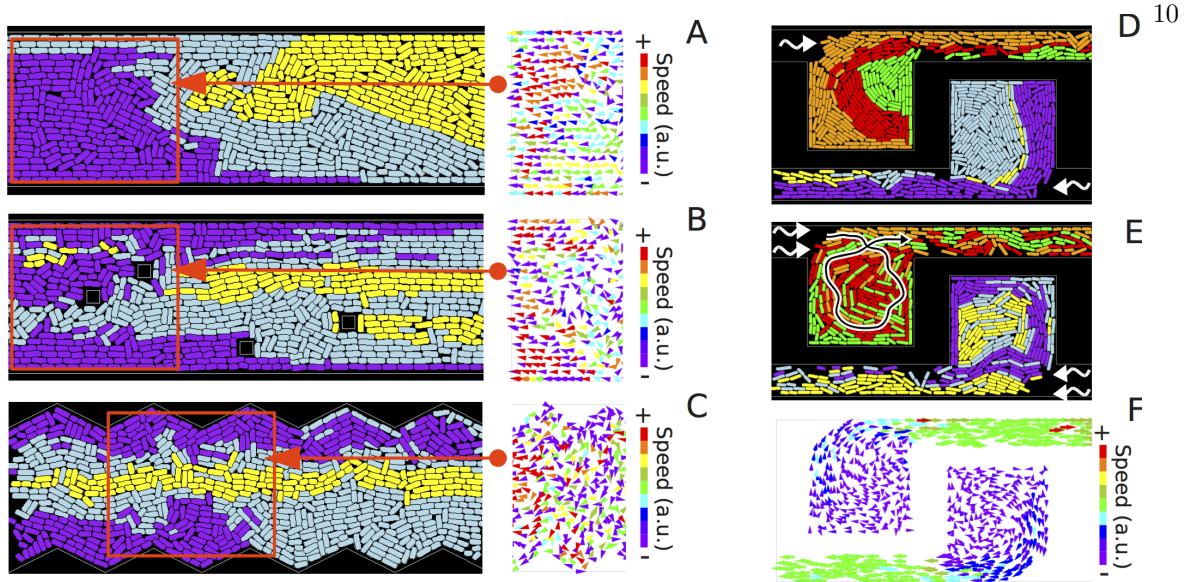


Figure 4. Dynamical mixing of bacterial strains under different conditions. (A): Three strains (purple, yellow and blue) growing in a longitudinal pipe. Detail (on the right) shows the vector field corresponding to the red square (on the left) where the arrows display both directionality and velocity of every cell. (B): Similar to (A), but with four *columns* placed in the middle of the pipe. (C): Similar to A, but with *zig-zag* borders along the pipe. In (A), (B) and (C), the speed in vector fields is measured in arbitrary units (a.u.). (D): Six strains (three per trap) grow in two square traps on one side of the main longitudinal channel. The flow in the channels follows the direction of the arrows. (E): Similar to (D) but with a much stronger flow in channels, causing turbulence in traps (long circled arrow). (F): Vector field of experiment (E).

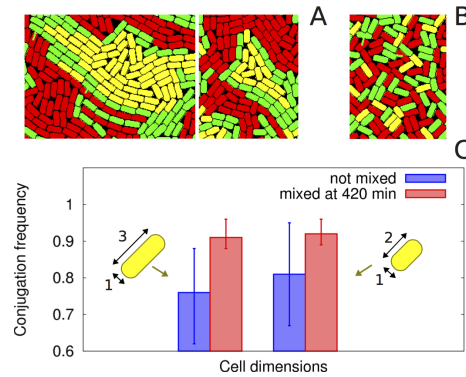


Figure 5. Effects of manual mixing on conjugation frequency. (A): Recipient-trapping behaviour of a population with donors (red), transconjugants (green) and recipients (yellow). Two snapshots depict clearly-observed clusters. (B): Population after random mixing, where the clusters are automatically dissolved. (C): Graph showing conjugation frequencies ($Y = T/(R + T)$) of 560-minute experiments (ratio D/R = 50%). Blue bars represent Y on an untouched population, while red bars represent Y when the population is mixed at 420 minutes. The two sets of bars correspond to experiments with different cell dimensions (1x3 -left- and 1x2 -right). Error bars show variation across 15 experiments of each class.

An interesting result from Figure 5C is the fact that the smaller the size of the cell, the higher results we observe for conjugation frequencies. This may be due to the fact that smaller cells are able to slip through physical gaps, and the biomechanical ordering of the population becomes more “fuzzy”. This underlines the importance of considering the physical shape of cells, since circle-shaped cells would not give valid results.

5 Discussion

The conjugation model presented here is the first agent-based model to explicitly simulate the conjugation process with growing rod-shaped cells. Full validation against real data is performed, which shows the capacity of the software to reproduce observed behaviour. In addition, the mixing study offers valuable insights into the design of multi-strain populations. The software also allows for genetic *programs* to be *installed* inside cells; the potential for horizontal gene transfer to recreate distributed information processing within a microbial consortium is of significant interest in synthetic biology, and the software presented will aid the design and testing of systems before their *wet-lab* implementation.

References

1. Heinemann M, Panke S (2006) Synthetic biology—putting engineering into biology. *Bioinformatics* 22: 2790–9.
2. Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. *Nature* 434: 1130–4.
3. Macía J, Posas F, Solé RV (2012) Distributed computation: the new wave of synthetic biology devices. *Trends in Biotechnology* 30: 342–9.
4. Tamsir A, Tabor JJ, Voigt CA (2011) Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. *Nature* 469: 212–5.
5. Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403: 339–42.
6. Ausländer S, Ausländer D, Müller M, Wieland M, Fussenegger M (2012) Programmable single-cell mammalian biocomputers. *Nature* 487: 123–127.
7. Atkinson S, Williams P (2009) Quorum sensing and social networking in the microbial world. *Journal of the Royal Society Interface* 6: 959–78.
8. Ortiz ME, Endy D (2012) Engineered cell-cell communication via DNA messaging. *Journal of Biological Engineering* 6.
9. de la Cruz F, Frost LS, Meyer RJ, Zechner EL (2010) Conjugative DNA metabolism in Gram-negative bacteria. *FEMS Microbiology Reviews* 34: 18–40.
10. del Campo I, Ruiz R, Cuevas A, Revilla C, Vielva L, et al. (2012) Determination of conjugation rates on solid surfaces. *Plasmid* 67: 174–82.
11. Gorochowski TE, Matyjaszkiewicz A, Todd T, Oak N, Kowalska K, et al. (2012) BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology. *PLOS ONE* 7: e42790.
12. Kreft JU, Booth G, Wimpenny JW (1998) BacSim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology* 144: 3275–87.
13. Melke P, Sahlin P, Levchenko A, Jo H (2010) A cell-based model for quorum sensing in heterogeneous bacterial colonies. *PLoS Computational Biology* 6: e1000819.
14. Rudge TJ, Steiner PJ, Phillips A, Haseloff J (2012) Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology* 1: 345–352.
15. Emonet T, Macal CM, North MJ, Wickersham CE, Cluzel P (2005) AgentCell: a digital single-cell assay for bacterial chemotaxis. *Bioinformatics* 21: 2714–21.

16. Izaguirre J, Chaturvedi R, Huang C, Cickovski T, Coffland J, et al. (2004) CompuCell, a multi-model framework for simulation of morphogenesis. *Bioinformatics* 20: 1129–37.
17. Holmes AB, Kalvala S, Whitworth DE (2010) Spatial simulations of myxobacterial development. *PLoS Computational Biology* 6: e1000686.
18. Krone SM, Lu R, Fox R, Suzuki H, Top EM (2007) Modelling the spatial dynamics of plasmid transfer and persistence. *Microbiology* 153: 2803–16.
19. Merkey BV, Lardon LA, Seoane JM, Kreft Ju, Smets BF (2011) Growth dependence of conjugation explains limited plasmid invasion in biofilms : an individual-based. *Environmental Microbiology* 13: 2435–2452.
20. Lardon L, Merkey B, Martins S, Dötsch A, Picioreanu C, et al. (2011) iDynoMiCS : next-generation individual-based modelling of biofilms. *Environmental Microbiology* 13: 2416–2434.
21. Seoane J, Yankelevich T, Dechesne A², Merkey B, Sternberg C, et al. (2011) An individual-based approach to explain plasmid invasion in bacterial populations. *FEMS Microbiology Ecology* 75: 17–27.
22. Guantes R, Poyatos JF (2006) Dynamical principles of two-component genetic oscillators. *PLoS Computational Biology* 2: e30.
23. Goni-Moreno A, Amos M (2012) A reconfigurable NAND/NOR genetic logic gate. *BMC Systems Biology* 6.
24. Volfson D, Cookson S, Hasty J, Tsimring LS (2008) Biomechanical ordering of dense cell populations. *Proceedings of the National Academy of Sciences of the United States of America* 105: 15346–51.
25. Garcillán-Barcia MP, Cruz FD (2008) Why is entry exclusion an essential feature of conjugative plasmids? *Plasmid* 60: 1–18.